Approximate Inference in Bayesian Networks

Outline

I. Direct sampling methods

II. Rejection sampling

III. Importance sampling

* Figures are either from the textbook site.
Approximate Inference Methods

- These methods, also called Monte Carlo algorithms, are based on the principle of stochastic sampling.
Approximate Inference Methods

- These methods, also called Monte Carlo algorithms, are based on the principle of stochastic sampling.
  - Generate random events based on the probabilities in the Bayes net.
  - Count up different answers found in these events.
  - Monte Carlo algorithms are used to estimate quantities that are difficult to calculate exactly.
Approximate Inference Methods

♦ These methods, also called Monte Carlo algorithms, are based on the principle of stochastic sampling.

- Generate random events based on the probabilities in the Bayes net.
- Count up different answers found in these events.
- Monte Carlo algorithms are used to estimate quantities that are difficult to calculate exactly.

♦ Accuracy depends on the size of the sample set.

- Can get arbitrarily close to the true probability distribution as the size increases.
Approximate Inference Methods

- These methods, also called Monte Carlo algorithms, are based on the principle of stochastic sampling.
  - Generate random events based on the probabilities in the Bayes net.
  - Count up different answers found in these events.
  - Monte Carlo algorithms are used to estimate quantities that are difficult to calculate exactly.

- Accuracy depends on the size of the sample set.
  - Can get arbitrarily close to the true probability distribution as the size increases.

- Two families of algorithms: direct sampling and Markov chain sampling.
I. Direct Sampling

• Sample variables in topological order.

The value of a node is *conditioned* on those already assigned to its parents (the assignments are *guaranteed* by the topological order).
I. Direct Sampling

- Sample variables in topological order.

  The value of a node is *conditioned* on those already assigned to its parents (the assignments are *guaranteed* by the topological order).

```plaintext
function PRIOR-SAMPLE(bn) returns an event sampled from the prior specified by bn
inputs: bn, a Bayesian network specifying joint distribution \( P(X_1, \ldots, X_n) \)

  \( x \leftarrow \) an event with \( n \) elements

for each variable \( X_i \) in \( X_1, \ldots, X_n \) do
  \( x[i] \leftarrow \) a random sample from \( P(X_i | \text{parents}(X_i)) \)

return \( x \)
```
I. Direct Sampling

- Sample variables in topological order.

The value of a node is *conditioned* on those already assigned to its parents (the assignments are *guaranteed* by the topological order).

```plaintext
function PRIOR-SAMPLE(bn) returns an event sampled from the prior specified by bn
inputs: bn, a Bayesian network specifying joint distribution \( P(X_1, \ldots, X_n) \)

\[
\begin{align*}
x & \leftarrow \text{an event with } n \text{ elements} \\
\text{for each } & \text{variable } X_i \text{ in } X_1, \ldots, X_n \text{ do} \\
 & \quad x[i] \leftarrow \text{a random sample from } P(X_i | \text{parents}(X_i)) \\
\text{return } & x \quad \text{from the domain of } X_i, \text{ e.g., true or false}
\end{align*}
```
The Sprinkler Network

- Every morning Mary checks the weather.
- If it’s cloudy, she usually does not turn on the sprinkler.
- The grass will be wet if the sprinkler is on, or if it rains during the day.
The Sprinkler Network

- Every morning Mary checks the weather.
- If it’s cloudy, she usually does not turn on the sprinkler.
- The grass will be wet if the sprinkler is on, or if it rains during the day.

Order: Cloudy, Sprinkler, Rain, WetGrass
The Sprinkler Network

- Every morning Mary checks the weather.
- If it’s cloudy, she usually does not turn on the sprinkler.
- The grass will be wet if the sprinkler is on, or if it rains during the day.

Order: Cloudy, Sprinkler, Rain, WetGrass

1. Value true sampled from $P(\text{Cloudy}) = \langle 0.5, 0.5 \rangle$. 
The Sprinkler Network

- Every morning Mary checks the weather.
- If it’s cloudy, she usually does not turn on the sprinkler.
- The grass will be wet if the sprinkler is on, or if it rains during the day.

Order: Cloudy, Sprinkler, Rain, WetGrass

1. Value true sampled from $P(\text{Cloudy}) = \langle 0.5, 0.5 \rangle$.

   How? Generate a pseudo-random number $r$ in the range $[0, 1]$. Return true if $r < 0.5$ and false otherwise.
The Sprinkler Network

- Every morning Mary checks the weather.
- If it’s cloudy, she usually does not turn on the sprinkler.
- The grass will be wet if the sprinkler is on, or if it rains during the day.

Order: Cloudy, Sprinkler, Rain, WetGrass

1. Value *true* sampled from $P(Cloudy) = \langle 0.5, 0.5 \rangle$.
2. Value *false* sampled from $P(Sprinkler | Cloudy = true) = \langle 0.1, 0.9 \rangle$.

How? Generate a pseudo-random number $r$ in the range $[0, 1]$. Return *true* if $r < 0.5$ and *false* otherwise.
The Sprinkler Network

- Every morning Mary checks the weather.
- If it’s cloudy, she usually does not turn on the sprinkler.
- The grass will be wet if the sprinkler is on, or if it rains during the day.

Order: *Cloudy, Sprinkler, Rain, WetGrass*

1. Value *true* sampled from $P(\text{Cloudy}) = \langle 0.5, 0.5 \rangle$.

2. Value *false* sampled from $P(\text{Sprinkler} \mid \text{Cloudy} = \text{true}) = \langle 0.1, 0.9 \rangle$.

3. Value *true* sampled from $P(\text{Rain} \mid \text{Cloudy} = \text{true}) = \langle 0.8, 0.2 \rangle$.
The Sprinkler Network

• Every morning Mary checks the weather.

• If it’s cloudy, she usually does not turn on the sprinkler.

• The grass will be wet if the sprinkler is on, or if it rains during the day.

Order: Cloudy, Sprinkler, Rain, WetGrass

1. Value \text{true} sampled from \( P(\text{Cloudy}) = \langle 0.5, 0.5 \rangle \).

2. Value \text{false} sampled from \( P(\text{Sprinkler} | \text{Cloudy} = \text{true}) = \langle 0.1, 0.9 \rangle \).

3. Value \text{true} sampled from \( P(\text{Rain} | \text{Cloudy} = \text{true}) = \langle 0.8, 0.2 \rangle \).

4. Value \text{true} sampled from \( P(\text{WetGrass} | \text{Sprinkler} = \text{false}, \text{Rain} = \text{true}) = \langle 0.9, 0.1 \rangle \).

How? Generate a pseudo-random number \( r \) in the range \([0, 1]\). Return true if \( r < 0.5 \) and false otherwise.
The Sprinkler Network

- Every morning Mary checks the weather.
- If it’s cloudy, she usually does not turn on the sprinkler.
- The grass will be wet if the sprinkler is on, or if it rains during the day.

Order: Cloudy, Sprinkler, Rain, WetGrass

1. Value true sampled from \( P(\text{Cloudy}) = \langle 0.5, 0.5 \rangle \).
2. Value false sampled from \( P(\text{Sprinkler} \mid \text{Cloudy} = \text{true}) = \langle 0.1, 0.9 \rangle \).
3. Value true sampled from \( P(\text{Rain} \mid \text{Cloudy} = \text{true}) = \langle 0.8, 0.2 \rangle \).
4. Value true sampled from \( P(\text{WetGrass} \mid \text{Sprinkler} = \text{false}, \text{Rain} = \text{true}) = \langle 0.9, 0.1 \rangle \).

Prior-Sample() returns the event [true, false, true, true].

How? Generate a pseudo-random number \( r \) in the range \([0, 1]\). Return true if \( r < 0.5 \) and false otherwise.
Sampling Process

$S_{PS}(x_1, ..., x_n)$: probability of a specific event generated by PRIOR-SAMPLE.
Sampling Process

\( S_{PS}(x_1, ..., x_n) \): probability of a specific event generated by PRIOR-SAMPLE.

\[
S_{PS}(x_1, ..., x_n) = \prod_{i=1}^{n} P(x_i \mid Parents(X_i))
\]
Sampling Process

\[ S_{PS}(x_1, \ldots, x_n) : \text{probability of a specific event generated by PRIOR-SAMPLE.} \]

\[
S_{PS}(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid Parents(X_i))
\]

\[
= P(x_1, \ldots, x_n) \quad \text{(probability of joint distribution)}
\]
Sampling Process

$S_{PS}(x_1, ..., x_n)$: probability of a specific event generated by PRIOR-SAMPLE.

$$S_{PS}(x_1, ..., x_n) = \prod_{i=1}^{n} P(x_i \mid Parents(X_i))$$

$$= P(x_1, ..., x_n) \quad \text{(probability of joint distribution)}$$

- $N$ total samples produced by PRIOR-SAMPLE.
Sampling Process

\[ S_{PS}(x_1, ..., x_n): \text{ probability of a specific event generated by } PRIOR\text{-SAMPLE.} \]

\[
S_{PS}(x_1, ..., x_n) = \prod_{i=1}^{n} P(x_i \mid Parents(X_i))
\]

\[
= P(x_1, ..., x_n) \quad \text{(probability of joint distribution)}
\]

- \( N \) total samples produced by \( PRIOR\text{-SAMPLE.} \)
- \( N_{ps}(x_1, ..., x_n) \) times that the specific event \( X_1 = x_1 \land ... \land X_n = x_n \) occurs.
Sampling Process

$S_{PS}(x_1, \ldots, x_n)$: probability of a specific event generated by PRIOR-SAMPLE.

$$S_{PS}(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid Parents(X_i))$$

$$= P(x_1, \ldots, x_n) \quad \text{(probability of joint distribution)}$$

- $N$ total samples produced by PRIOR-SAMPLE.
- $N_{ps}(x_1, \ldots, x_n)$ times that the specific event $X_1 = x_1 \land \cdots \land X_n = x_n$ occurs.

$$\lim_{N \to \infty} \frac{N_{ps}(x_1, \ldots, x_n)}{N} = S_{PS}(x_1, \ldots, x_n) = P(x_1, \ldots, x_n)$$
Sampling Process

\[ S_{PS}(x_1, \ldots, x_n): \text{probability of a specific event generated by PRIOR-SAMPLE.} \]

\[
S_{PS}(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid \text{Parents}(X_i))
\]

\[
= P(x_1, \ldots, x_n) \quad \text{(probability of joint distribution)}
\]

- \( N \) total samples produced by PRIOR-SAMPLE.
- \( N_{ps}(x_1, \ldots, x_n) \) times that the specific event \( X_1 = x_1 \land \cdots \land X_n = x_n \) occurs.

\[
\lim_{N \to \infty} \frac{N_{ps}(x_1, \ldots, x_n)}{N} = S_{PS}(x_1, \ldots, x_n) = P(x_1, \ldots, x_n)
\]

\[ S_{PS}(\text{true, false, true, true}) = 0.5 \cdot 0.9 \cdot 0.8 \cdot 0.9 = 0.324 \]
Partially Specified Event

Estimate the probability of the partial event $X_1 = x_1 \land \cdots \land X_m = x_m$, $m \leq n$:

$$P(x_1, \ldots, x_m) \approx \frac{N_{ps}(x_1, \ldots, x_m)}{N} \equiv \hat{P}(x_1, \ldots, x_m)$$

the fraction of all sampled complete events that match the partially specified event
Partially Specified Event

Estimate the probability of the partial event $X_1 = x_1 \land \cdots \land X_m = x_m, m \leq n$:

$$P(x_1, \ldots, x_m) \approx \frac{N_{ps}(x_1, \ldots, x_m)}{N} \equiv \hat{P}(x_1, \ldots, x_m)$$

the fraction of all sampled complete events that match the partially specified event.

Example $Rain = true$ holds for 511 of 1,000 samples generated from the sprinkler network.

$$\hat{P}(Rain = true) = 0.511$$
II. Rejection Sampling (RS)

Generate samples from a hard-to-sample distribution given an easy-to-sample distribution.

Estimate $P(X \mid e)$ as follows:
II. Rejection Sampling (RS)

Generate samples from a hard-to-sample distribution given an easy-to-sample distribution.

Estimate $P(X \mid e)$ as follows:

- Generate samples from the prior distribution specified by the BN.
II. Rejection Sampling (RS)

Generate samples from a hard-to-sample distribution given an easy-to-sample distribution.

Estimate $P(X | e)$ as follows:

- Generate samples from the prior distribution specified by the BN.
- Rejects all those that do not match the evidence $e$.

$N_{PS}(e)$: the number of samples that are not rejected.
II. Rejection Sampling (RS)

Generate samples from a hard-to-sample distribution given an easy-to-sample distribution.

Estimate $P(X \mid e)$ as follows:

- Generate samples from the prior distribution specified by the BN.
- Rejects all those that do not match the evidence $e$.

$N_{PS}(e)$: the number of samples that are not rejected.

- Count how often $X = x$ occurs in the remaining $N_{PS}(e)$ samples for every value $x$ of $X$. 

II. Rejection Sampling (RS)

Generate samples from a hard-to-sample distribution given an easy-to-sample distribution.

Estimate $P(X \mid e)$ as follows:

- Generate samples from the prior distribution specified by the BN.
- Rejects all those that do not match the evidence $e$.

$N_{PS}(e)$: the number of samples that are not rejected.

- Count how often $X = x$ occurs in the remaining $N_{PS}(e)$ samples for every value $x$ of $X$.

$N_{PS}(X, e)$: the vector of counts, one for each value $x$ of $X$, of samples that contain $X = x$ and also agree with the evidence $e$. 
II. Rejection Sampling (RS)

Generate samples from a hard-to-sample distribution given an easy-to-sample distribution.

Estimate $P(X \mid e)$ as follows:

- Generate samples from the prior distribution specified by the BN.
- Rejects all those that do not match the evidence $e$.

\[ N_{PS}(e): \text{the number of samples that are not rejected.} \]

- Count how often $X = x$ occurs in the remaining $N_{PS}(e)$ samples for every value $x$ of $X$.

\[ N_{PS}(X, e): \text{the vector of counts, one for each value } x \text{ of } X, \text{of samples that contain } X = x \text{ and also agree with the evidence } e. \]

\[ \hat{P}(X \mid e) = \alpha N_{PS}(X, e) = \frac{N_{PS}(X, e)}{N_{PS}(e)} \]
Estimation of True Probability by RS

\[
\hat{P}(X \mid e) = \frac{N_{PS}(X, e)}{N_{PS}(e)}
\]
Estimation of True Probability by RS

\[ \hat{P}(X \mid e) = \frac{N_{PS}(X, e)}{N_{PS}(e)} \]

\[ = \frac{N_{PS}(X, e)/N}{N_{PS}(e)/N} \]
Estimation of True Probability by RS

\[ \hat{P}(X \mid e) = \frac{N_{PS}(X, e)}{N_{PS}(e)} \]

\[ = \frac{N_{PS}(X, e)/N}{N_{PS}(e)/N} \]

\[ \approx \frac{P(X, e)}{P(e)} \]

since \( P(x_1, ..., x_m) \approx \frac{N_{ps}(x_1, ..., x_m)}{N} \)
Estimation of True Probability by RS

\[ \hat{P}(X | e) = \frac{N_{PS}(X, e)}{N_{PS}(e)} = \frac{N_{PS}(X, e)/N}{N_{PS}(e)/N} \]

\[ \approx \frac{P(X, e)}{P(e)} \]

since \( P(x_1, ..., x_m) \approx \frac{N_{ps}(x_1, ..., x_m)}{N} \)

\[ \hat{P}(X | e) \approx P(X | e) \]
Estimation of True Probability by RS

\[
\hat{P}(X | e) = \frac{N_{PS}(X, e)}{N_{PS}(e)} = \frac{N_{PS}(X, e)/N}{N_{PS}(e)/N}
\]

Standard deviation \(1/\sqrt{N}\) of error in each probability

\[
\hat{P}(X | e) \approx \frac{P(X, e)}{P(e)}
\]

since \(P(x_1, ..., x_m) \approx \frac{N_{ps}(x_1, ..., x_m)}{N}\)
Estimation of True Probability by RS

\[ \hat{P}(X \mid e) = \frac{N_{PS}(X, e)}{N_{PS}(e)} \]

\[ = \frac{N_{PS}(X, e) / N}{N_{PS}(e) / N} \]

Standard deviation \(1/\sqrt{N}\) of error in each probability

\[ \hat{P}(X \mid e) \approx P(X \mid e) \]

Example: Estimate \(P(Rain \mid Sprinkler = true)\) using 100 samples.
Estimation of True Probability by RS

\[ \hat{P}(X | e) = \frac{N_{PS}(X, e)}{N_{PS}(e)} = \frac{N_{PS}(X, e)/N}{N_{PS}(e)/N} \]

Standard deviation \(1/\sqrt{N}\) of error in each probability

\[ \hat{P}(X | e) \approx \frac{P(X, e)}{P(e)} \]

Example Estimate \(P(Rain | Sprinkler = true)\) using 100 samples.

- 73 samples have \(Sprinkler = false\) and are rejected.
- Of the 27 samples with \(Sprinkler = true\), only 8 have \(Rain = true\).
Estimation of True Probability by RS

\[ \hat{P}(X \mid e) = \frac{N_{PS}(X, e)}{N_{PS}(e)} = \frac{N_{PS}(X, e)/N}{N_{PS}(e)/N} \]

Standard deviation \(\frac{1}{\sqrt{N}}\) of error in each probability

\[ \approx \frac{P(X, e)}{P(e)} \] since \(P(x_1, \ldots, x_m) \approx \frac{N_{ps}(x_1, \ldots, x_m)}{N} \)

Example Estimate \(P(Rain \mid Sprinkler = true)\) using 100 samples.

- 73 samples have \(Sprinkler = false\) and are rejected.
- Of the 27 samples with \(Sprinkler = true\), only 8 have \(Rain = true\).

\[ P(Rain \mid Sprinkler = true) \approx \alpha(8,19) = (0.296,0.704) \]
How Fast Does RS Converge?

- How many samples are needed before the resulting estimates are close to the correct answers with high probability?

- The complexity of rejection sampling depends primarily on the fraction of samples that are accepted.

  \[ \text{prior probability of the evidence } P(e) \]
How Fast Does RS Converge?

♦ How many samples are needed before the resulting estimates are close to the correct answers with high probability?

♦ The complexity of rejection sampling depends primarily on the fraction of samples that are accepted.

\[ \text{fraction of samples accepted} = \frac{P(e)}{P(e)} \]

= prior probability of the evidence \( P(e) \)

♦ \( P(e) \) is vanishingly small for complex networks with many evidence variables.

• The fraction of samples consistent with \( e \) drops exponentially as the number of evidence variables grows.

• Rejection sampling is unusable for complex problems.
III. Importance Sampling (IS)

♦ Emulate the effect of sampling from one distribution $P$ using samples from another distribution $Q$.

✦ It is too hard to sample from the true posterior distribution on all the evidence.

✦ So we sample from an easy distribution.

♦ To ensure correctness in the limit, we use a correction factor $P(x)/Q(x)$, to each sample $x$ when it is counted.
III. Importance Sampling (IS)

- Emulate the effect of sampling from one distribution $P$ using samples from another distribution $Q$.
  - It is too hard to sample from the true posterior distribution on all the evidence.
  - So we sample from an easy distribution.

- To ensure correctness in the limit, we use a correction factor $P(x)/Q(x)$, to each sample $x$ when it is counted.

How does this work?
Query $P(Rain \mid Cloudy = true, WetGrass = true)$
Weight of a Sample

Query \( P(Rain \mid Cloudy = true, WetGrass = true) \)

- Pick a topological order:

  Cloudy, Sprinkler, Rain, WetGrass
Weight of a Sample

Query \( P(Rain \mid Cloudy = true, WetGrass = true) \)

- Pick a topological order:
  - Cloudy, Sprinkler, Rain, WetGrass

Evidence variables \( E = e \)
Weight of a Sample

Query $P(Rain \mid Cloudy = true, WetGrass = true)$

- Pick a topological order:

  Cloudy, Sprinkler, Rain, WetGrass
**Weight of a Sample**

**Query** $P(Rain \mid Cloudy = true, WetGrass = true)$

- Pick a topological order:

```
Cloudy, Sprinkler, Rain, WetGrass
```

<table>
<thead>
<tr>
<th>$C$</th>
<th>$P(S\mid c)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$</td>
<td>.10</td>
</tr>
<tr>
<td>$f$</td>
<td>.50</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$S$</th>
<th>$R$</th>
<th>$P(W\mid s,r)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$</td>
<td>$t$</td>
<td>.99</td>
</tr>
<tr>
<td>$t$</td>
<td>$f$</td>
<td>.90</td>
</tr>
<tr>
<td>$f$</td>
<td>$t$</td>
<td>.90</td>
</tr>
<tr>
<td>$f$</td>
<td>$f$</td>
<td>.00</td>
</tr>
</tbody>
</table>
Weight of a Sample

Query $P(Rain \mid Cloudy = true, WetGrass = true)$

- Pick a topological order:
  
  $Cloudy, Sprinkler, Rain, WetGrass$

- Set the weight $w = 1$. 
(cont’d)

- Generate an event in the chosen topological order.
• Generate an event in the chosen topological order.

1. *Cloudy* is an evidence variable with value *true*.

\[ w \leftarrow w \times P(Cloudy = true) = 0.5 \]
• Generate an event in the chosen topological order.

1. *Cloudy* is an evidence variable with value *true*.
   
   \[ w \leftarrow w \times P(\text{Cloudy} = \text{true}) = 0.5 \]

2. *Sprinkler* is not an evidence variable. Sample from \( P(\text{Sprinkler} | \text{Cloudy} = \text{true}) = \langle 0.1, 0.9 \rangle \).
• Generate an event in the chosen topological order.

1. *Cloudy* is an evidence variable with value *true*.
   
   \[ w \leftarrow w \times P(\text{Cloudy} = \text{true}) = 0.5 \]

2. *Sprinkler* is not an evidence variable. Sample from \[ P(\text{Sprinkler} | \text{Cloudy} = \text{true}) = \langle 0.1, 0.9 \rangle. \]
• Generate an event in the chosen topological order.

1. *Cloudy* is an evidence variable with value *true*.

   \[ w \leftarrow w \times P(Cloudy = \text{true}) = 0.5 \]

2. *Sprinkler* is not an evidence variable. Sample from

   \[ P(Sprinkler \mid Cloudy = \text{true}) = \langle 0.1, 0.9 \rangle. \]

   Suppose this returns *false*. (cont’d)
• Generate an event in the chosen topological order.

1. *Cloudy* is an evidence variable with value *true*.
   
   \[ w \leftarrow w \times P(\text{Cloudy} = \text{true}) = 0.5 \]

2. *Sprinkler* is not an evidence variable. Sample from 
   \[ P(\text{Sprinkler} \mid \text{Cloudy} = \text{true}) = \langle 0.1, 0.9 \rangle. \]
   
   Suppose this returns *false*.

3. *Rain* is not an evidence variable. Sample from 
   \[ P(\text{Rain} \mid \text{Cloudy} = \text{true}) = \langle 0.8, 0.2 \rangle. \]
• Generate an event in the chosen topological order.

  1. *Cloudy* is an evidence variable with value *true*.
     
     \[ w \leftarrow w \times P(\text{Cloudy} = \text{true}) = 0.5 \]

  2. *Sprinkler* is not an evidence variable. Sample from \( P(\text{Sprinkler} | \text{Cloudy} = \text{true}) = \langle 0.1, 0.9 \rangle \).
     
     Suppose this returns *false*.

  3. *Rain* is not an evidence variable. Sample from \( P(\text{Rain} | \text{Cloudy} = \text{true}) = \langle 0.8, 0.2 \rangle \).
• Generate an event in the chosen topological order.
  1. *Cloudy* is an evidence variable with value *true*.
     \[ w \leftarrow w \times P(\text{Cloudy} = \text{true}) = 0.5 \]
  2. *Sprinkler* is not an evidence variable. Sample from 
     \[ P(Sprinkler \mid Cloudy = \text{true}) = \langle 0.1, 0.9 \rangle. \]
     Suppose this returns *false*.
  3. *Rain* is not an evidence variable. Sample from 
     \[ P(Rain \mid Cloudy = \text{true}) = \langle 0.8, 0.2 \rangle. \]
     Suppose this returns *true*. 
• Generate an event in the chosen topological order.

1. *Cloudy* is an evidence variable with value *true*.
   \[ w \leftarrow w \times P(Cloudy = true) = 0.5 \]

2. *Sprinkler* is not an evidence variable. Sample from
   \[ P(Sprinkler \mid Cloudy = true) = \langle 0.1, 0.9 \rangle. \]
   Suppose this returns *false*.

3. *Rain* is not an evidence variable. Sample from
   \[ P(Rain \mid Cloudy = true) = \langle 0.8, 0.2 \rangle. \]
   Suppose this returns *true*.

4. *WetGrass* is an evidence variable with value *true*. 

• Generate an event in the chosen topological order.

1. *Cloudy* is an evidence variable with value *true*.
   
   \[ w \leftarrow w \times P(Cloudy = true) = 0.5 \]

2. *Sprinkler* is not an evidence variable. Sample from \( P(Sprinkler \mid Cloudy = true) = \langle 0.1, 0.9 \rangle \).
   
   Suppose this returns *false*.

3. *Rain* is not an evidence variable. Sample from \( P(Rain \mid Cloudy = true) = \langle 0.8, 0.2 \rangle \).
   
   Suppose this returns *true*.

4. *WetGrass* is an evidence variable with value *true*.  

Generate an event in the chosen topological order.

1. *Cloudy* is an evidence variable with value *true*.
   
   \[ w \leftarrow w \times P(\text{Cloudy} = \text{true}) = 0.5 \]

2. *Sprinkler* is not an evidence variable. Sample from \( P(\text{Sprinkler} | \text{Cloudy} = \text{true}) = (0.1,0.9) \).
   
   Suppose this returns *false*.

3. *Rain* is not an evidence variable. Sample from \( P(\text{Rain} | \text{Cloudy} = \text{true}) = (0.8,0.2) \).
   
   Suppose this returns *true*.

4. *WetGrass* is an evidence variable with value *true*.
   
   \[ w \leftarrow w \times P(\text{WetGrass} = \text{true} | \text{Sprinkler} = \text{false}, \text{Rain} = \text{true}) = 0.5 \times 0.9 = 0.45 \]
• Generate an event in the chosen topological order.

1. *Cloudy* is an evidence variable with value *true*.
   
   \[ w \leftarrow w \times P(Cloudy = true) = 0.5 \]

2. *Sprinkler* is not an evidence variable. Sample from 
   \[ P(Sprinkler | Cloudy = true) = \langle 0.1, 0.9 \rangle. \]
   
   Suppose this returns *false*.

3. *Rain* is not an evidence variable. Sample from 
   \[ P(Rain | Cloudy = true) = \langle 0.8, 0.2 \rangle. \]
   
   Suppose this returns *true*.

4. *WetGrass* is an evidence variable with value *true*.
   
   \[ w \leftarrow w \times P(WetGrass = true | Sprinkler = false, Rain = true) = 0.5 \times 0.9 = 0.45 \]

• This round of sampling returns the event \([true, false, true, true]\) with weight 0.45.
(cont’d)

• Generate an event in the chosen topological order.

1. *Cloudy* is an evidence variable with value *true*.
   
   \[ \text{\( w \leftarrow w \times P(\text{Cloudy} = \text{true}) = 0.5 \)} \]

2. *Sprinkler* is not an evidence variable. Sample from 
   \( P(\text{Sprinkler} | \text{Cloudy} = \text{true}) = \langle 0.1, 0.9 \rangle \).
   
   Suppose this returns *false*.

3. *Rain* is not an evidence variable. Sample from 
   \( P(\text{Rain} | \text{Cloudy} = \text{true}) = \langle 0.8, 0.2 \rangle \).
   
   Suppose this returns *true*.

4. *WetGrass* is an evidence variable with value *true*.
   
   \[ \text{\( w \leftarrow w \times P(\text{WetGrass} = \text{true} | \text{Sprinkler} = \text{false}, \text{Rain} = \text{true}) = 0.45 \)} \]

• This round of sampling returns the event \([\text{true, false, true, true}]\) with weight 0.45.

• This event is tallied under \( \text{Rain} = \text{true} \) in generating the distribution estimate 
  \( \hat{P}(\text{Rain} | \text{Cloudy} = \text{true}, \text{WetGrass} = \text{true}) \).
Likelihood Weighting

*Fix* the values for the evidence variables $E$ and sample all the nonevidence variables $\{X\} \cup Z$ in topological order, each conditioned on its parents.
Likelihood Weighting

Fix the values for the evidence variables $E$ and sample all the nonevidence variables $\{X\} \cup Z$ in topological order, each conditioned on its parents.

Sampling distribution of the hidden variables $Z = \{Z_1, \ldots, Z_l\}$ (for evidence $E = e$):

$$Q_{WS}(z) = \prod_{i=1}^{l} P(z_i) \mid \text{parents}(Z_i))$$
**Likelihood Weighting**

*Fix* the values for the evidence variables $E$ and sample all the nonevidence variables $\{X\} \cup Z$ in topological order, each conditioned on its parents.

Sampling distribution of the hidden variables $Z = \{Z_1, \ldots, Z_l\}$ (for evidence $E = e$):

$$Q_{WS}(z) = \prod_{i=1}^{l} P(z_i | \text{parents}(Z_i))$$

The weight $w(z)$ must satisfy

$$P(z, e) = P(e)P(z | e) = P(e)w(z)Q_{WS}(z)$$
Likelihood Weighting

**Fix** the values for the evidence variables $E$ and sample all the nonevidence variables $\{X\} \cup Z$ in topological order, each conditioned on its parents.

Sampling distribution of the hidden variables $Z = \{Z_1, \ldots, Z_l\}$ (for evidence $E = e$):

$$Q_{WS}(z) = \prod_{i=1}^{l} P(z_i \mid \text{parents}(Z_i))$$

The weight $w(z)$ must satisfy

$$P(z, e) = P(e)P(z \mid e) = P(e)w(z)Q_{WS}(z)$$

$$w(z) = \frac{1}{P(e)} \cdot \frac{P(z, e)}{Q_{WS}(z)} = \alpha \frac{P(z, e)}{Q_{WS}(z)}$$
Likelihood Weighting

Fix the values for the evidence variables $E$ and sample all the nonevidence variables $\{X\} \cup Z$ in topological order, each conditioned on its parents.

Sampling distribution of the hidden variables $Z = \{Z_1, \ldots, Z_l\}$ (for evidence $E = e$):

$$Q_{WS}(z) = \prod_{i=1}^{l} P(z_i) \mid \text{parents}(Z_i))$$

The weight $w(z)$ must satisfy

$$P(z, e) = P(e)P(z \mid e) = P(e)w(z)Q_{WS}(z)$$

$$w(z) = \frac{1}{P(e)} \cdot \frac{P(z, e)}{Q_{WS}(z)} = \alpha \frac{P(z, e)}{Q_{WS}(z)}$$

(normalization factor $\alpha = 1/P(e)$)
Likelihood Weighting

Fix the values for the evidence variables $E$ and sample all the nonevidence variables $\{X\} \cup Z$ in topological order, each conditioned on its parents.

Sampling distribution of the hidden variables $Z = \{Z_1, \ldots, Z_l\}$ (for evidence $E = e$):

$$Q_{WS}(z) = \prod_{i=1}^{l} P(z_i) \mid parents(Z_i))$$

The weight $w(z)$ must satisfy

$$P(z, e) = P(e)P(z \mid e) = P(e)w(z)Q_{WS}(z)$$

$$w(z) = \frac{1}{P(e)} \cdot \frac{P(z, e)}{Q_{WS}(z)} = \frac{P(z, e)}{Q_{WS}(z)}$$

(normalization factor $\alpha = 1/P(e)$)

$$= \alpha \frac{\prod_{i=1}^{l} P(z_i) \mid parents(Z_i)) \cdot \prod_{i=1}^{m} P(e_i) \mid parents(E_i))}{\prod_{i=1}^{l} P(z_i) \mid parents(Z_i))}$$
Likelihood Weighting

Fix the values for the evidence variables $E$ and sample all the nonevidence variables $\{X\} \cup Z$ in topological order, each conditioned on its parents.

Sampling distribution of the hidden variables $Z = \{Z_1, ..., Z_l\}$ (for evidence $E = e$):

$$Q_{WS}(z) = \prod_{i=1}^{l} P(z_i) \mid \text{parents}(Z_i))$$

The weight $w(z)$ must satisfy

$$P(z, e) = P(e)P(z \mid e) = P(e)w(z)Q_{WS}(z)$$

$$w(z) = \frac{1}{P(e)} \cdot \frac{P(z,e)}{Q_{WS}(z)} = \alpha \frac{P(z,e)}{Q_{WS}(z)} \quad \text{(normalization factor } \alpha = 1/P(e))$$

$$= \alpha \frac{\prod_{i=1}^{l} P(z_i) \mid \text{parents}(Z_i)) \cdot \prod_{i=1}^{m} P(e_i) \mid \text{parents}(E_i))}{\prod_{i=1}^{l} P(z_i) \mid \text{parents}(Z_i))}$$

$$= \alpha \prod_{i=1}^{m} P(e_i) \mid \text{parents}(E_i))$$
Weighted Sampling

\[ w(z) = \alpha \prod_{i=1}^{m} P(e_i | \text{parents}(E_i)) \]

The weight is the product of the conditional probabilities for the evidence variables given their parents.

```
function WEIGHTED-SAMPLE(bn, e) returns an event and a weight
    w ← 1; x ← an event with n elements, with values fixed from e
    for i = 1 to n do
        if \( X_i \) is an evidence variable with value \( x_{ij} \) in e
            then w ← w \times P(X_i = x_{ij} | \text{parents}(X_i))
        else x[i] ← a random sample from \( P(X_i | \text{parents}(X_i)) \)
    return x, w
```
The Likelihood Weighting Algorithm

function \textsc{Likelihood-Weighting}(X, e, bn, N) returns an estimate of \( P(X \mid e) \)
inputs: \( X \), the query variable
\( e \), observed values for variables \( E \)
\( bn \), a Bayesian network specifying joint distribution \( P(X_1, \ldots, X_n) \)
\( N \), the total number of samples to be generated
local variables: \( W \), a vector of weighted counts for each value of \( X \), initially zero

for \( j = 1 \) to \( N \) do
\( x, w \leftarrow \textsc{Weighted-Sample}(bn, e) \)
\( W[j] \leftarrow W[j] + w \) where \( x_j \) is the value of \( X \) in \( x \)
return \( \textsc{Normalize}(W) \)

function \textsc{Weighted-Sample}(bn, e) returns an event and a weight
\( w \leftarrow 1; x \leftarrow \) an event with \( n \) elements, with values fixed from \( e \)
for \( i = 1 \) to \( n \) do
if \( X_i \) is an evidence variable with value \( x_{ij} \) in \( e \)
then \( w \leftarrow w \times P(X_i = x_{ij} \mid \text{parents}(X_i)) \)
else \( x[i] \leftarrow \) a random sample from \( P(X_i \mid \text{parents}(X_i)) \)
return \( x, w \)
Performance Comparison

On the car insurance network

Likelihood weighting is considerably more efficient than rejection sampling.